



Středoškolská technika 2012

Setkání a prezentace prací středoškolských studentů na ČVUT

Ovládání automatické frézky

Petr Tomášek

Střední průmyslová škola Hronov
Hostovského 910, Hronov

Úvod

Cílem tohoto projektu bylo vytvoření programů na zakázku, které budou ovládat tří osou CNC frézku s třemi krokovými motory, přes universální desku. Zařízení musí být možné ovládat jak ručně, pomocí předního panelu (**Mikroprocesor – ruční ovládání**), tak pomocí počítače, který bude s frézku komunikovat pomocí sériového rozhraní USB (**Počítač – ruční ovládání, Počítač – automatické ovládání**). V zařízení je tedy umístěn mikroprocesor, který tuto komunikace obstarává. Jednotlivé pohyby motorů musí být signalizovány pomocí LED diod.

Popis funkce

Projekt byl uskutečněn na přání p. uč. Linharta, kterému se podařilo sehnat kostru frézky, se třemi krokovými motory. Představa vypadala asi takto: Jakmile bude zprovozněna elektronická část zařízení, bude třeba vymyslet způsob, kterým lze řídit ovládání frézky pomocí ručních tlačítek přímo na panelu frézky, pomocí ručního ovládání z počítače a v poslední fázi projektu musí být grézka schopna vykreslit obrazec vytvořený pomocí programu SprintLayout (nebo jiný, který podporuje GERBER export).

Ovládací panel frézky je tvořen pomocí **šesti tlačítek** (jednotlivé směry pohybu ve třech osách: „X, Y, Z“), **dvou dvoupolohových přepínačů** (rychlost pohybu: Pomalu - frekvence **100Hz** / Rychle - frekvence **1KHz**, přepínání ovládání: Ručně - ovládací panel zařízení / Automaticky - ovládání pomocí počítače) a **indikačních diod** pro pohyb frézky a zapnutí zařízení. Musíme uvažovat, že pro každý motor jsou potřeba dvě tlačítka. Pro kladný a záporný pohyb po směrové ose. Celkem tedy včetně již zmíněných přepínačů máme 8 vstupů. Indikace pohybu některého z motoru je realizována pomocí LED diody, která je aktivní v logické „1“ (úroveň **High**). V každém směru může být stisknuto pouze jedno tlačítko

pohybu (jedno aktivní pro každou osu). Pokud by se ke dvěma tlačítkům přidalo jakékoli třetí (nebo více), přístroj bude vykonávat stejný pohyb, který byl nastaven na výstupu, před stiskem následujícího tlačítka (na výstupu zůstane předchozí stav). Přístroj tedy dokáže vykreslit úhly i 45 stupňů. Řízení z PC umožní změnu rychlostí pohybu a tím i jinou velikost úhlů.

Jestliže je přepínač Automaticky/Ručně přepnut na polohu automaticky (úroveň **Low**), tak jednotlivá tlačítka na panelu frézky nesmí reagovat. Naopak, pokud je tentýž vypínač přepnut do logické hodnoty „1“ (úroveň **High**), nesmí mikroprocesor přijímat data z počítače. Realizována je také signalizace jednotlivých pohybů motorů, pomocí samostatných nízko-příkonových LED diod, umístěných na výstupech mikroprocesoru. Jakmile je vybuzen jeden z výstupů, dioda se rozsvítí. (tyto diody jsou aktivní v úrovni **Low**). Diody slouží k signalizaci aktivních kombinací úrovní na výstupu mikroprocesoru, což se využije zejména k ladění aplikace.

Aby bylo možné uskutečnit tento projekt, bylo třeba si zvolit, pomocí jakého způsobu řízení bude zařízení fungovat. Jelikož bylo zapotřebí ovládat zařízení také přes počítač, vybráno bylo řízení pomocí mikroprocesoru AT89S52 od firmy Atmel.

Začneme tedy od začátku. Pomocí univerzální desky pro řízení tří krokových motorů by bylo možné připojit přímo Sériový (RS232) nebo Paralelní port (LPT – „Line Printer Terminal“) počítače. Bohužel v dnešní době jen málo notebooků je opatřeno Sériovým nebo dokonce Paralelním portem. Proto bylo zapotřebí vyřešit převod mezi RS232 a USB. K tomu byl vybrán integrovaný obvod s označením FT232 od firmy FTDI, který nám převede signál z počítače do mikroprocesoru. Použitý mikroprocesor pracuje s TTL úrovněmi stejně jako FTDI převodník => Nebylo zapotřebí řešit převody mezi logickými úrovněmi (například IO Max232).

Ruční ovládání:

Jak již bylo zmíněno, zařízení má pro ruční ovládání celkem **8 vstupů**. Přepínání: Pomalu/Rychle je vyřešeno změnou hodnot v použitých časovačích procesoru. Při přepnutí na „Pomalou“ je nastavena frekvence na **100Hz**. Naopak, pokud přepneme na „Rychle“, frekvence se pohybuje na **1Khz**. Jednotlivá tlačítka pro pohyb motorů jsou aktivní v logické „0“.

Automatické ovládání:

Komunikace mezi mikroprocesorem a počítačem je obstarávána pomocí 10 odesílaných Bajtů z počítače, přičemž poslední Bajt je vždy kontrolní součet devíti předchozích. Každou setinu sekundy odesíláme do mikroprocesoru právě těchto 10 Bajtů. Odesílání dělíme na test komunikace se zařízením a odesílání jednotlivých dat pro požadovaný pohyb. Celá komunikace je řízena dvěma linkami **TxD** a **RxD (+ GND)**, tyto linky zajišťují veškerou komunikaci s počítačem. Program pro automatické ovládání je napsán v Jazyce **C++ Builder**.

Fotografie hotového zařízení

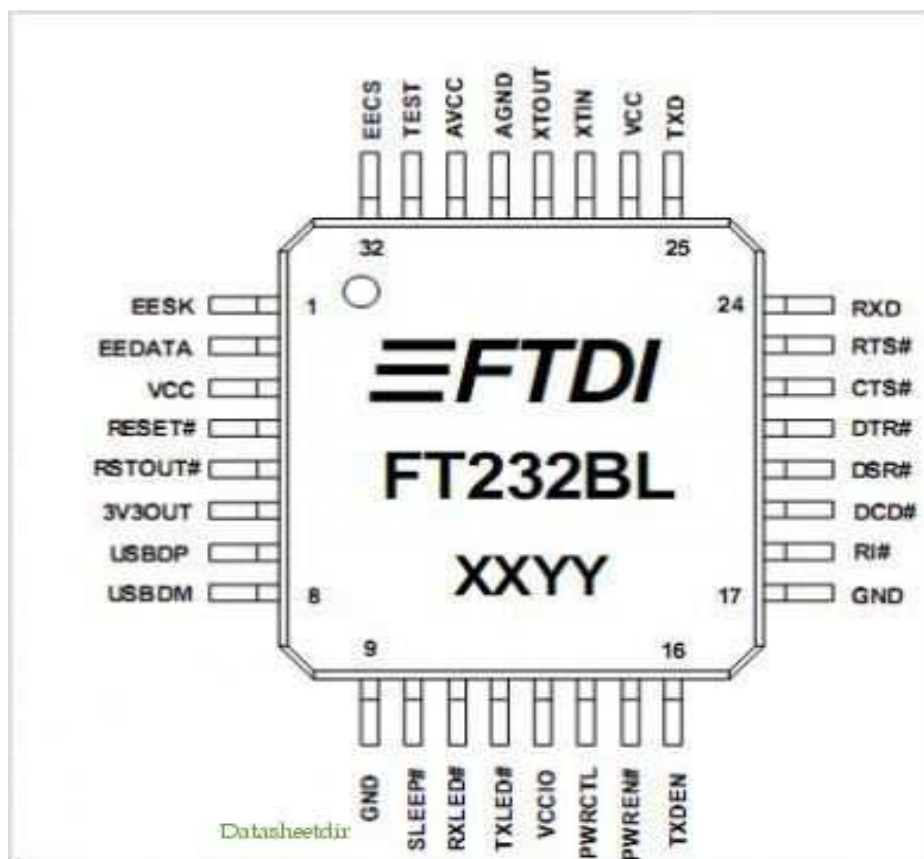


Komunikace s FT232

Obvody od firmy FTDI se dají rozdělit do tří skupin.

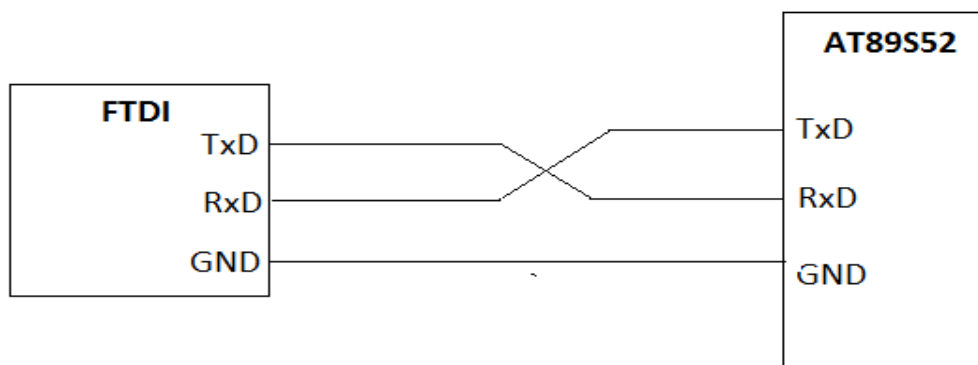
1. USB – UART (Universal Asynchronous Receiver Transmitter), které pracují se Sériovým rozhraním
2. USB – FIFO (first in, first out), které pracují s 8b paralelním rozhraním
3. USB - UART/FIFO, které pracují s oběma rozhraními.

V tomto zařízení je použit první typ IO USB-UART, tedy ten, který disponuje sériovým rozhraním. USB-UART podporuje 7/8 datových bitů, 1/2 stop bity a několik druhů parity. Přenosová rychlost obvodu se pohybuje od 300bit/s – 3Mb/s. Je zde možnost plného řízení přenosu, signály: RTS#, CTS#, DTR#, DSR#, DCD#, RI#. Pozn.: Všechny tyto vývody jsou INVERTOVANÉ! IO dále umožňuje práci s 5V a 3.3V logikou, přičemž samotný obvod lze napájet přímo z USB. K obvodu lze připojit Externí paměť EEPROM, do které se dá uložit název zařízení, pod kterým se poté zařízení po připojení přihlásí.



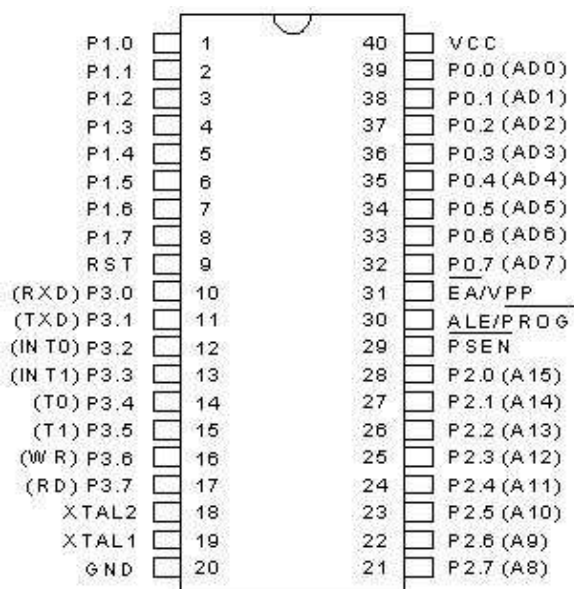
TxD (Transmit Data) – vysílání dat
RxD (Received Data) – příjem dat

Jedná se o dva vodiče, pomocí kterých probíhá komunikace mezi zařízením a počítačem přes USB-UART. Aby komunikace mohla fungovat, je zapotřebí přidat ještě třetí vodič, tentokrát společný, kterým je zem (Ground nebo GND).



Mikroprocesor - seznámení s AT89S52

Jedná se o CMOS 8b mikroprocesor s programovatelnou 4KB systémovou Flash pamětí a 128B pamětí RAM. Umožňuje až 5 zdrojů přerušení. Obsahuje 32 programovatelných I/O linek (port P0 – P3), 2 x 16b časovače nebo čítače. Mikroprocesor obsahuje také SPI (Serial Peripheral Interface) rozhraní. V tomto zařízení je použit krystal 11.0592Mhz.



Mikroprocesor - ruční ovládání:

První část programu pokud nebereme v úvahu komentáře začíná tzv. „hlavičkou“, klíčovým slovem \$MOD, které říká překladači, pro jaký typ procesoru je program napsán. V našem případě se jedná o typ 52.

Druhou část programu slouží k přiřazení (EQU), vytvoření pomocných bitů v paměti (BIT) a vytvoření pomocných bytů v paměti (DATA).

Třetí částí programu je Organizace paměti (ORG). Jakmile je program spuštěn, tak je proveden „skok“ v našem případě na adresu 00H s návěštím NINI. V organizaci jsou také udávány adresy všech přerušení smyčky programu.

Čtvrtá část programu se nazývá Inicializace. Slouží k nastavení počátečních hodnot proměnných, registrů, bitů, časovačů, výstupů apod.

Pátá část slouží pro ošetření tzv. „zákmitů“, které vznikají při sepnutí nebo uvolnění tlačítka. V tomto okamžiku dochází k „odskočení“ kontaktu, což vede k vytvoření několika sestupných a náběžných hran. Tím pádem dochází k vytvoření více impulsů místo jednoho a tomuto případu se říká zákmit.

Šestá část programu realizuje logické rovnice pro jednotlivé výstupy.

Například pro výstup OMXI:

```
MOV C,PTXP
ANL C,/PTXM
MOV VYPO,C
MOV C,PTXM
ANL C,/PTXP
ORL C,VYPO
ANL C,GIMP
MOV OMXI,C
```

1. Pomocí funkce MOV je načtena do příznaku přenosu (Carry) hodnota předchozího stavu vstupu TXP.
2. Je realizován logický součin přenosu s negovaným předchozím stavem vstupu TXM.
3. Výsledek je uložen do pomocného bitu VYPO.
4. Opět načtena předchozí hodnota do přenosu Carry, tentokrát vstupu TXM
5. Realizace logického součinu Carry s negovaným předchozím stavem TXP.
6. Realizovaný logický součet přenosu Carry s pomocným bitem VYPO
7. V předposledním kroku je realizován logický součin přenosu Carry s pomocným bitem generátor impulsů. (GIMP).
8. Poslední řádek posílá výslednou rovnici na výstup OMXI.

Sedmá část provede návrat na začátek hlavní smyčky programu.

Mikroprocesor - automatické ovládání:

V předchozích kapitolách jsem se zmínil o odesílání 10B z počítače do mikroprocesoru. A tímto také začneme. Pokud nám mikroprocesor přijme data z počítače, jsou uloženy na jednotlivé adresy v paměti procesoru a také je prováděn jejich kontrolní součet. Přesněji řečeno prvních 9 bytů (V 10. bytu je uložen KS, který vypočítal program v C++ builderu). Data jsou tedy přijmuta a součet devíti bytů odpovídá 10. přijmutému bytu. Jestliže je přijmuto všech 10 bytů a KS souhlasí, program se větví na různé stavy, podle hodnoty prvního přijmutého bytu:

01 – kontrolní součet

02 – komunikace RA

...

Podle těchto hodnot je program dále větven na jednotlivé reakce pohybů, popřípadě jak již bylo zmíněno kontrolní součet, který pošle stav zařízení. Automatické ovládání je nyní rozděleno na dvě různé části, přičemž obě jsou vytvořeny v jediné aplikaci pomocí C++ Builderu:

1) Automatické + ruční (odpovídající panel na frézce je vytvořen také v počítači)

2) Automatické + Automatické (frézka sama vykresluje námi nakreslený obrazec)

- 1) Část „AR“ funguje podobně jako předchozí řízení „ručně“, s rozdílem, že jsou nejdříve vždy do pomocných bitů sériových hodnot (př.: STXP) načteny hodnoty z pomocného registru B. (SDATA). Poté jsou realizovány stejné logické rovnice, jako u ručního ovládání, s rozdílem, že předchozí stavy vstupů po přečtení tlačítka automaticky/ručně (z důvodu náhlého přepnutí druhu ovládání) jsou nahrazeny sériovými hodnotami jednotlivých vstupů. (př.: STXP).
- 2) Část „AA“ funguje trochu jiným způsobem. Po přijmutí 10 bajtů jsou jednotlivé hodnoty počtů impulsů a rychlosti načteny do paměti mikroprocesoru (v tu chvíli je již přijímáno dalších 10B, pokud jsou připraveny, aby se urychlila celková činnost frézky) a s každým přetečením časovače (pro šikmé pohyby využívány dva časovače s různými rychlostmi) jsou tyto hodnoty dekrementovány a porovnávány s nulou => (pohyb dokončen). Tento cyklus se opakuje do té doby, dokud přichází z počítače data a pokud souhlasí jejich kontrolní součet.

Seznámení s C++ Builder

Jedná se o vývojové prostředí od firmy Borland, která vydala například podobné prostředí **Delphi**. Pokud porovnáme **klasické** programování (sekvenční), které se používalo dříve, například **DOS**, **Pascal**, tak je zřejmé, že tvorba aplikací ve Vizuelním prostředí C++ Builder je ohromně zjednodušená. Jednotlivé funkce, které zabíraly dříve několik řádků, se v C++ Builder dají napsat pár slovy, na jeden řádek. Dále pomocí **vizuálního programování** nám program pomáhá psát i zdrojový kód. Pokud začneme psát určitý příkaz, automaticky se vyvolá nápověda, která nám nabídne slova, ve kterých je obsažena námi zapsaná část příkazu.

C++ Builder oproti „čistému“ C se liší také tzv. OOP neboli **objektově orientované programování**. Tj. programovací styl, který používá oddělené objekty, které v sobě nesou svá data a zdrojový kód. => každý objekt má své místo v určité třídě a stejně tak, každý **objekt za svá data zodpovídá**.

Počítač - ruční ovládání:

První část programu tvoří tzv. „Direktivy“. Jsou to instrukce přímo pro překladač. Direktiva **#INCLUDE** například vkládá soubory (z pravidla hlavičkové) do zdrojového kódu.

Druhou část programu tvoří Konstruktor (implicitní) a Destruktor. Konstruktor je volán při vytváření objektu, naopak destruktory při zániku objektu, v našem případě jej používáme pro ukončení USB komunikace.

Třetí částí programu jsou vlastní funkce. Každý pohyb jednotlivých motorů má svoji vlastní volanou funkci pomocí **__fastcall** a je třídy **TfrmHlavni**. Pokud tedy je stisknut pohyb doleva, je volána funkce **Yminus()**. Každá z těchto funkcí může odesílat dvě různé hodnoty Bytu (pro pomalý a rychlý pohyb frézky), za předpokladu, že je zařízení připojeno a nenastal-li problém v komunikaci.

Příklad pro pohyb Xplus():

Pro pomalý pohyb je odeslána hodnota: FB hexadecimálně.

Pro rychlý pohyb je odeslána hodnota: F9 hexadecimálně.

Čtvrtá část programu obsahuje funkce volané při různých událostech. Mezi nimi je i funkce **FormCreate()**, třídy **TfrmHlavni**.

V této funkci dochází tak po úspěšném otevření zařízení s názvem „Frezka“ k nastavení komunikace s UBS sběrnici. Zajišťují to tyto tři řádky.:

```
FT_SetBaudRate(ftHandle, 9600);
FT_SetTimeouts(ftHandle, 100, 100);
FT_SetDataCharacteristics(ftHandle, FT_BITS_8, FT_STOP_BITS_1, FT_PARITY_NONE);
```

První řádek nám nastavuje přenosovou rychlost komunikace v baudech [Bd]. Druhý řádek poté nastaví „čas čekání“ při čtení a zápisu. Nakonec třetí řádek nastavuje počet datových bitů, počet stop-bitů a paritu.

Pátá část programu volá funkci FormKeyDown(.....&Key), která mimo jiných obsahuje ukazatel na parametr Key, ve kterém je uložena hodnota Ascii kódu tlačítka, které bylo stisknuto. Je tedy volána pokaždé, když je stisknuto tlačítko na klávesnici.

Šestou a zároveň poslední část programu tvoří zbylé funkce, které jsou patří k položkám menu, případně k rychlému přístupu.

Počítač - automatické ovládání:

Část „AA“ je o dost komplikovanější. Pokud vezmeme v úvahu fakt, že frézka musí sama vykreslovat určitý obrazec, který si my jako uživatelé nakreslíme, není to úplně snadná záležitost. Jak jsem ji zmiňoval, projekt byl vytvořen hlavně pro vytváření popiskových štítků, upozorňovacích tabulek apod. K tomu nám bohatě stačí program, který je ve škole používám pro tvorbu plošných spojů. Tento program dokáže generovat tzv. gerber formát, což není nic jiného, než souřadnice jednotlivých bodů námi nakresleného obrazce. Tyto souřadnice je potřeba zpracovat a odeslat.

Zpracování probíhá ve zkratce následovně:

- a) Procházíme jednotlivé řádky načteného soboru.
- b) Kontrolujeme hodnotu aktuálního řádku + řádku následujícího a vyhodnotíme danou situaci. (rytí frézky, pohyb nad objektem, posun nahoru a dolů).
- c) Na jednotlivé situace voláme příslušné funkce, pro zpracování požadavků, ve kterých nejprve vypočítáme rychlost, pomocí Pythagorovi věty a vzorce uvedeného ve zdrojovém kódu pro mikroprocesor pro výpočet rychlosti. Vzorec byl odvozen od použité frekvence oscilátoru, maximální hodnoty časovače/čítače v mikroprocesoru a konstanty určené krokovými motory + universální deskou.
- d) Souřadnice, které jsme obdrželi v parametrech již zmíněných funkcí pro zpracování daného pohybu (aktuální řádek, následující řádek) od sebe odečítáme, získáváme jednotlivé výsledné pohyby, konkrétněji přesný počet impulsů potřebný pro daný pohyb. Kladná a záporná hodnota výsledku nám značí pohyb kladný, popřípadě záporný.
- e) Tyto výsledky jsou převáděny společně s vypočítanou rychlostí do šestnáctkové soustavy (hexadecimální) a odesílány přes FTDI převodník jako data do mikroprocesoru.

Použitá literatura – internetové zdroje

<http://www.keil.com/dd/docs/datashts/atmel/>

<http://www.usconverters.com/downloads/>

Použitá literatura – knihy

Učíme se programovat v Borland C++ Builder